IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:   Not Yet Assigned

Filing Date: Herewith

Applicant: Patrick Urban

Group Art Unit: Not Yet Assigned

Examiner: Not Yet Assigned

Title: DISPLAY CONTROL WITH ACTIVE HYPERTEXT
                     DOCUMENTS

Attorney Docket: 7347-000001

Hon. Commissioner of Patents and Trademarks
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Sir:

Prior to the examination of this application, please amend it as follows:

IN THE SPECIFICATION

Please replace the following paragraphs of the specification. Applicant includes herewith an Attachment for Specification Amendments showing a marked up version of each replacement paragraph.

[Page 1, lines 9-16] In the application of data processing, a distinction is drawn between the processing logic, i.e. the actual program, and the interaction with the user via a user interface, usually given this name, UI, or called a graphical user interface,

(GUI). A known example is the GUI of the Windows operating systems, which is integrated into the executable program via interface calls. In this case, it is not possible to separate the processing logic from the user interface calls. That is to say the processing logic and the structure of the interface are integrally combined with one another.

[Page 4, lines 28-34 and Page 5, lines 1-5]   The event monitoring mechanism **56** of the show1.htm page is an "onChange" JavaScript command in the present example. The "onChange" command monitors the input field generated by the user interaction input mechanism. When the JavaScript detects a change event (such as, when the user focus leaves a field into which text was typed) the event handler mechanism sends an event message to the JavaScript **40**. In this example, the event message is a "putText" command that is sent back to the JavaScript **40** of the controlling page **26**. It bears noting that the "onChange" event command and the "putText" event message are merely examples, suitable for capturing text input from the user. Any event detection mechanisms and event messages available in the executable scripting language may be used to effect the desired results.

[Page 6, lines 13-22]   Listing 2 shows the HTML text "show1.htm" loaded into the frame in which interaction with the user takes place. In this simple example, this merely comprises the prompt "Please Enter Text" in the second line and an input field in line 4, the "onChange" parameter in line 5 defining JAVASCRIPT event handling for said input field. This stipulates that the "putText" function defined in line 4 of Listing 1 be called

2

using the content of the input field designated by the "this.value" construct, when the field

content is changed. Depending on the browser, this function occurs when the TAB key or

the ENTER key is used to leave the field, possibly depending on the further content of the

page.

[Page 7, lines 3-10] Execution of Line 7 in Listing 1 loads the HTML text shown in

Listing 3 into the frame in which "show1.htm" has been active up to now. This screen

comprises a prompt "Your Input:" in line 4, a field "was" used for the output in line 5 and a

button "Again" in line 6. The "onload" statement in line 2 calls the "getText" function, again

from the superordinate frameset in Listing 1, in that case lines 8-9, and stores the result in

the output field "was" in line 5 Listing 3. Once this HTML page has loaded, the previously

entered text thus appears in the input field.

[Page 9, line 12]   11.   top.show.location="show1.htm";}

IN THE CLAIMS

Please amend the claims in accordance with the following rewritten claims

in clean form. Applicant includes herewith an Attachment for Claim Amendments

showing a marked up version of each amended claim.

Claim 9. (AMENDED)   A method of using a computer browser for interpreting

user input through a user interface page that is separate from a sequence controller

page comprising:

creating at least one user interface page with a markup language;

embedding executable code for an event handling mechanism in the user interface page;

display at least one user interface page;

creating a separate sequence controller page with a markup language;

embedding executable code for a sequence control program in the sequence controller page that will respond to even messages received through the displayed user interface page;

displaying at least one user interface page as a result of the event message received by the sequence control program; and

repeating the process of the sequence controller program in responding to event messages received through the currently displayed user interface page while event messages are received.

Claim 11. (AMENDED)   The method of claim 10 wherein the sequence controller page is without variable content.

Claim 12. (AMENDED)   The method of claim 9 further comprising splitting a display area into frames including one frame that contains the sequence controller page without variable content.

Claim 15. (AMENDED)   An apparatus for a computer based browser to interpret user input through a user interface page that is separate from a sequence controller page comprising:

4

a displayed user interface page produced by a markup language;

an event handling mechanism created by executed code embedded in the user interface page that will receive user input;

a separate sequence controller page produced by a markup language; and

a sequence control program created by executable code embedded in the sequence controller page that responds to the event handling mechanism.

Claim 17. (AMENDED)   The apparatus of claim 16 wherein the sequence controller page is without variable content.

Claim 19. (AMENDED)   The apparatus of claim 15 further comprising a browser capable of splitting a display area into frames including one frame that contains the sequence controller page without variable content.

Please add the following new claims.

Claim 24. (NEW)   The method as claimed in claim 3, where a display program for the markup language provides for the display area to be split into frames, and a frame without variable contents is used for the sequence controller page.

REMARKS

The purpose of this preliminary amendment is to clarify the application for purposes unrelated to patentability and to add new claims.
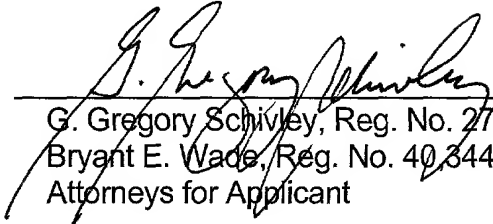
Favorable consideration of this application is respectfully requested.

Respectfully submitted,

Date: April ___6___ , 2001

By: _____
G. Gregory Schivley, Reg. No. 27,382
Bryant E. Wade, Reg. No. 40,344
Attorneys for Applicant

Harness, Dickey & Pierce, P.L.C.
P.O. Box 828
Bloomfield Hills, MI 48303
(248) 641-1600
GGS/BEW/msm

6

ATTACHMENT FOR SPECIFICATION AMENDMENTS

The following is a marked up version of each replacement paragraph and/or section of the specification in which underlines indicates insertions and brackets indicate deletions.

[Page 1, lines 9-16] In [one] the application of data processing, a distinction is drawn between the processing logic, i.e. the actual program, and the interaction with the user via a user interface, usually given this name, UI, or called a graphical user interface, GUI. A known example is the GUI of the Windows operating systems, which is integrated into the executable program via interface calls. In this case, it is not possible to separate the processing logic from the user interface calls. That is to say the processing logic and the structure of the interface are integrally combined with one another.

[Page 4, lines 28-34 and Page 5, lines 1-5]   The event monitoring mechanism **56** of the show1.htm page is an "onChange" JavaScript command in the present example. The "onChange" command monitors the input field generated by the user interaction input mechanism. When the JavaScript detects a change event (such as, when the user focus leaves a field into which text was typed [types text into the field] the event handler mechanism sends an event message to the JavaScript **40**. In this example, the event message is a "putText" command that is sent back to the JavaScript **40** of the controlling page **26**. It bears noting that the "onChange" event command and the "putText" event

message are merely examples, suitable for capturing text input from the user. Any event detection mechanisms and event messages available in the executable scripting language may be used to effect the desired results.

[Page 6, lines 13-22]   Listing 2 shows the HTML text "show1.htm" loaded into the frame in which interaction with the user takes place. In this simple example, this merely comprises the prompt "Please Enter Text" in the second line and an input field in line 4, the "onChange" parameter in line 5 defining JAVASCRIPT event handling for said input field. This stipulates that the "putText" function defined in line 4 of Listing 1 be called using the content of the ["this.value"] input field <u>designated by the "this.value" construct</u>, when the field content is changed. Depending on the browser, this function occurs when the TAB key or the ENTER key is used to leave the field, possibly depending on the further content of the page.

[Page 7, lines 3-10] Execution of Line [5] <u>7</u> in Listing 1 loads the HTML text shown in Listing 3 into the frame in which "show1.htm" has been active up to now. This screen comprises a prompt "Your Input:" in line 4, a field "was" used for the output in line 5 and a button "Again" in line 6. The "onload" statement in line 2 calls the "getText" function, again from the superordinate frameset in Listing 1, in that case lines 8-9, and stores the result in the output field "was" in line 5 Listing 3. Once this HTML page has loaded, the previously entered text thus appears in the input field.

[Page 9, line 12]   11.   [-] top.show.location="show1.htm";}

8

ATTACHMENT FOR CLAIM AMENDMENTS

The following is a marked up version of each amended claim in which underlines indicates insertions and brackets indicate deletions.

Claim 9. (AMENDED)   A method of using a computer browser for interpreting user input through a user interface page that is separate from a sequence controller page comprising:

creating at least one user interface page with a markup language;

embedding executable code for an event handling mechanism in the user interface page;

display at least one user interface page;

creating a separate sequence controller page with a markup language;

embedding executable code for a sequence control program in the sequence controller page that will respond to even messages received through the displayed user interface page;

displaying at least one user interface page as a result of the event message received by the sequence control program; and

repeating the process of the sequence controller program in responding to event messages received through the currently displayed user interface page while event messages are received.

9

Claim 11. (AMENDED)   The method of claim 10 wherein the sequence controller page [user display] is without variable content.

Claim 12. (AMENDED)   The method of claim 9 further comprising splitting a display area into frames including one frame that contains the sequence controller page without variable content.

Claim 15. (AMENDED)   An apparatus for a computer based browser to interpret user input through a user interface page that is separate from a sequence controller page comprising:

a displayed user interface page produced by a markup language;

an event handling mechanism created by executed code embedded in the user interface page that will receive user input;

a separate sequence controller page produced by a markup language; and

a sequence control program created by executable code embedded in the sequence controller page that responds to the event handling mechanism[:].

Claim 17. (AMENDED)   The apparatus of claim 16 wherein the sequence controller page [user display] is without variable content.

Claim 19. (AMENDED)   The apparatus of claim 15 further comprising a browser capable of splitting a display area into frames including one frame that contains the sequence controller page without variable content.

10